EXHIBIT A

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

SYNOPSYS, INC., a Delaware corporation, Plaintiff, v. MAGMA DESIGN AUTOMATION, INC., a Delaware corporation, Defendant.	)	C.A. No. 05-701 (GMS)

DECLARATION OF DAVID HARRIS, PH.D. IN SUPPORT OF SYNOPSYS, INC.'S OPENING CLAIM CONSTRUCTION BRIEF

I, DAVID HARRIS, declare:

1. I have been retained as an expert consultant by the law firm of Dechert LLP ("Dechert"), counsel of record for plaintiff Synopsys, Inc. ("Synopsys") in the above-captioned matter. The following declaration is based on my personal knowledge. If called upon to testify, I could testify competently as to the matters set forth herein.

I. QUALIFICATIONS.

2. I received a Ph.D. in electrical engineering in 1999 from Stanford University in Stanford, California. I received several degrees from the Massachusetts Institute of Technology in 1994 including a Master's degree in Electrical Engineering and Computer Science, and Bachelor of Science degrees in Electrical Engineering and Mathematics.

3. I have over 15 years of experience in computer science and electrical engineering within industrial and academic settings. Over the years, I have received a number of different honors including: National Science Foundation fellow; IEEE Senior Membership; Irwin Sizer Award for Most Significant Improvement to MIT Education; and Master of Engineering Thesis Prize recipient. I am named as an inventor on eleven issued and several pending patents, all in fields relating to electrical engineering and computer science. I have authored three books and over thirty publications in the fields of electrical engineering and computer science. I am a co-author of a textbook on Logical Effort, which is a mathematical theory that has been used since its inception in 1991 to solve a wide variety of different problems arising in the design of integrated circuits. I recently finished a text book on digital circuit design and computer architecture, which is due to be published in February, 2007.

4. I have been teaching courses in electrical engineering and computer science at various universities and corporations for over twelve years. Currently, I am a professor of

electrical engineering at Harvey Mudd College in Claremont, California.

5. I am involved in teaching and research in the field of Integrated Circuit ("IC") Design. In the course of my teaching and research, I often analyze problems associated with logic design for achieving various objectives including timing objectives and area objectives.

6. After receiving my Master's degree in 1994, I worked for Intel Corporation as an IC designer of high-speed microprocessors. My primary contributions to these projects included the design and development of high-speed circuits, and these efforts led to the application and issuance of four U.S. patents. Since leaving Intel in 1997, I have been employed as a visiting professor at Sun Microsystems Labs in Menlo Park, California, where I have been studying fast circuits.

7. Throughout my professional career, I have used electronic design automation ("EDA") tools provided by many different manufacturers, including Synopsys, to design high-speed ICs. In the course of my doctoral research at Stanford and later research at Harvey Mudd, I developed a number of EDA tools. For example, I developed a timing analyzer that accounts for non-uniform clock skew. Also, I developed an EDA tool for performing library analysis of standard cell libraries.

8. My resume, attached as Exhibit A, describes my other work experience as well as provides details on my other qualifications.

II. REQUISITE LEVEL OF SKILL IN THE ART.

9. This declaration is submitted in support of Synopsys' Opening Claim Construction Brief regarding construction of the claims of U.S. Patent No. 6,192,508 ("the '508 patent") and U.S. Patent No. 6,519,745 ("the '745 patent"). Copies of the '508 and '745 patents are found in the Joint Appendix of Intrinsic and Extrinsic Evidence (the "JA") at (1 A-1 - A-9)

and (8 A-56 - A-69), respectively.[1]

10. The subject matter described in the `508 and `745 patents relates generally to the field of software design. The operation and design of ICs evoke scientific principles in the fields of computer science and electrical engineering. These principles collectively define a subject of endeavor referred to as the "art." In light of my qualifications, I believe I am an expert in the art. I believe a person of ordinary skill in the art would have either: a Bachelor of Science degree in computer science or electrical engineering with 2-5 yrs experience in software design; or a Master of Science degree in computer science or electrical engineering with 1-3 yrs experience in software design.

III. MATERIALS CONSIDERED.

11. In preparation for this declaration, I have reviewed the specification, claims and file history of the `508 and `745 Patents. I have also reviewed the Joint Claim Chart ("JCC") submitted by the parties on October 13, 2006, including materials cited therein. In addition, I have reviewed the Amended Joint Claim Chart ("AJCC") to be submitted by the parties on November 3, 2006, including materials cited therein.

IV. BACKGROUND OF TECHNOLOGY OF THE PATENTS AT ISSUE.

12. An IC is a miniature electronic circuit fabricated on a silicon substrate. People use ICs, also commonly known as "computer chips," on a daily basis in electronics equipment such as telephones, cameras, computers and automobiles.

13. Before an IC can be fabricated in a manufacturing facility, it must be designed to

---

[1] Pursuant to D. Del. LR 7.1.3(a)(C), all citations are to the Joint Appendix of Intrinsic and Extrinsic Evidence (the "JA"). Citations are in the following format: "[Tab#] A-[page#]." Thus, a citation to "3 A-100" would refer to tab 3, page A-100. Patent citations are by column and line number, in the following format: "[Tab#] A-[page#] at [column]:[line]."

ensure that it will serve the purpose for which it is intended. To this end, engineers must initially create a description of an IC, which is called a "layout." An IC layout will guide the ultimate fabrication of an IC in the same way that a blueprint guides the construction of a building.

14. Like architects, IC designers use many specialized tools in their trade. In the early days of the electronic industry, IC designers used pencil and paper to sketch out what the circuit would look like. IC's have become so complex, however, that designers now need powerful computer software to design a circuit layout. In fact, the process of designing a single IC has become so complex that it is divided into many stages, in which the designers must use different kinds of computer software to complete various tasks.

15. An entire industry, called the electronic design automation ("EDA") industry, is dedicated to developing software products (called "EDA Tools") that facilitate the various stages of the IC design process. Synopsys and Magma are both vendors of EDA Tools.

A. IC TERMINOLOGY.

16. A modern IC includes millions of IC elements formed on a substrate (or "chip"). The most common IC element is a gate (or "cell"). A gate is a small electronic circuit that performs a logical function such as comparing two electronic signals. Gates are interconnected with each other by wires to form an IC, according to the specified layout. Figure A, below, shows a portion of an IC including three gates A, B, and C connected by wires.

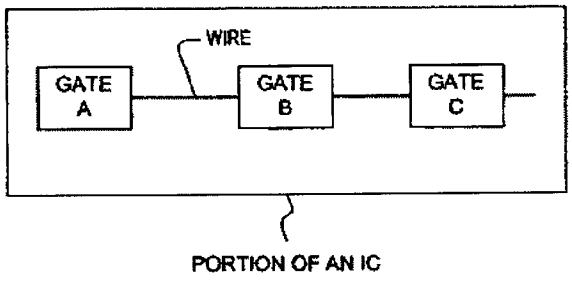

Figure A

B. EDA TERMINOLOGY.

17. As mentioned, IC designers use many different kinds of EDA Tools. Designers use these tools for various tasks, which are commonly performed in discrete stages that make up an IC "design flow." Early in the design flow, designers consider the intended purpose of the IC, and then begin the task of defining functions that the IC will need to perform. After the functions are specified, the designer can turn to the task of determining what types of IC elements (*e.g.*, what types of gates) should be used, and how the gates should be interconnected to achieve the specified functions. Later, the designers must address the task of determining where to place the gates, and how to route the wires between the gates. As described below, these tasks generally outline the major stages of the IC design flow.

1. Creating A High Level Circuit Description.

18. Early in the design flow, a designer creates a high-level circuit description using a coded language (*e.g.*, register transfer language). This description is called "high-level" because it only specifies what the circuit needs to do (*i.e.*, the function of the circuit), without describing what gates are to be used or how they should be connected. As an example, consider the design of a circuit for controlling operations in a cell phone. A high-level circuit description for such a circuit might contain the following description, expressed in Equation (1), below:

notify_user=(ring_on OR vibrate_on) AND power_on AND incoming_call (1)

19. The above equation describes a circuit operative to cause a cell phone to notify a user of an incoming call (*i.e.*, either by ringing or vibrating) whenever the cell phone power is turned on and an incoming call is received. While this description specifies what the circuit must do, many additional tasks must be performed to design a layout for an IC that could perform this function.

2. Logic Synthesis.

20. Figure B shows the basic function of a logic synthesis tool, which is to transform a high level circuit (*e.g.*, the description in equation (1), above) into a somewhat more detailed circuit description, called a "netlist."

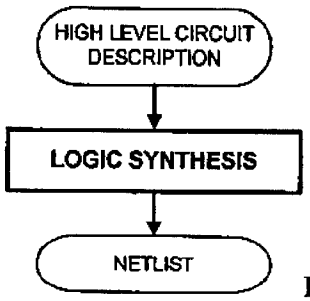

Figure B

21. During the "logic synthesis" stage, the high-level circuit description is transformed into a "netlist." A netlist contains a list of cells, and their interconnections. Referring back to the example of a cell phone, the high-level description of equation (1) includes several logical operations, including an "OR" operation and "AND" operations. During the logic synthesis stage, the high-level description of equation (1) may be transformed into a netlist, depicted by the following circuit diagram.

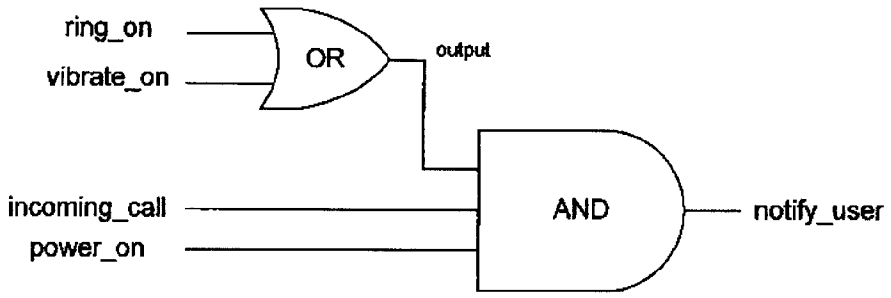

Figure C

22. Referring to the above diagram, the output of the "OR" gate provides a signal indicating "true" (*e.g.*, logical "1") if the ring_on signal is true "or" the vibrate_on signal is true. The output of the "AND" gate asserts the notify_user signal as "true" if the output of the "OR"

gate is true "and" the incoming_call signal is true "and" the power_on signal is true. Thus, the circuit shown in Figure C will assert the notify_user signal as true in accordance with the high-level description of equation (1). The circuit shown in Figure C is a graphical depiction of a "netlist" because it shows two types of gates (*i.e.*, a two-input OR gate, and a three-input AND gate) as well as the interconnection between the gates (*e.g.*, the output of the OR gate is connected to the first input of the AND gate). Apart from the graphical depiction, a netlist is simply a description of cells and their interconnections.

3. Placement.

23. Physical design refers to the process of placing and routing gates on an IC to create a "layout," which is a detailed blueprint describing the actual physical dimensions of the circuit elements (*e.g.*, gates and wires) to be formed on the chip. At the end of the design process, the layout may be used to manufacture the chip. The placement step, depicted generally in Figure D1, below, includes determining physical locations on the chip for all cells in the netlist.

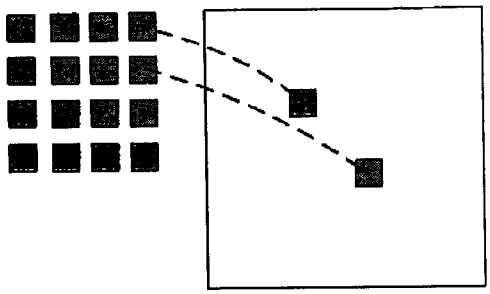

Figure D1

24. Placement is an iterative process wherein each cell is initially placed in a first physical location, and later moved to a different revised location. Since at least the mid 1990's, a number of different EDA companies have offered commercially available placement tools. There are many well known algorithms for performing placement.

25. In accordance with one common algorithm, the chip is divided up into "buckets"

(*i.e.*, portions of the area of the chip) prior to placement. Thereafter, each cell is initially placed into one of the buckets. Figure D2, below, shows the general idea of placing cells in buckets.

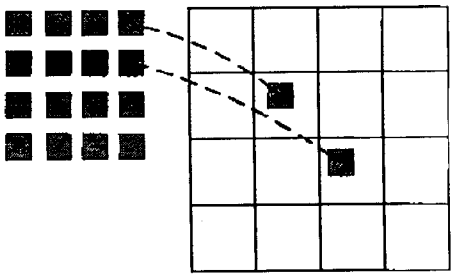

Figure D2

26. In accordance with another algorithm, cells may be placed in accordance with a process known as "partitioning." Partitioning generally refers to a process of dividing the area of the chip and performing operations within the divided areas. As is well known to persons of ordinary skill in the art, placement is typically an iterative process regardless of whether or not it involves partitioning. If a partitioning algorithm is employed in the placement process, the partitioning may be performed once or over multiple stages of an iterative placement process. In the example of Figure D2, the chip area is "partitioned" once into the bins shown in Figure D2. However, in an iterative partitioning process, each of the bins shown in Figure D2 may be further partitioned into smaller bins, and those smaller bins may again be partitioned into yet smaller bins and so on. This repeated partitioning could be performed over the course multiple stages of the placement process.

4. Routing

27. The routing step, depicted in Figure E below, includes determining the routes of the wires connecting the cells.

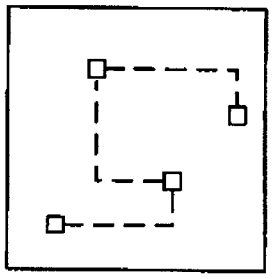

Figure E

28. The routing step becomes very challenging as the number of cells increases, and as the area of the chip decreases. As explained below, this situation leads to increased congestion, which makes routing very difficult because there is less space to lay down the wires.

C. SIGNIFICANT CHALLENGES IN THE EDA INDUSTRY

29. Industry requirements dictate that ICs be small, yet powerful and fast. Over the past three decades, technology advances in the semiconductor industry have dramatically increased the size, speed, and capacity of ICs. These ever-improving ICs can only be designed with sophisticated EDA tools. These tools are necessary to reduce the time and effort required to design chips, improve the performance and density of complex IC designs, and enhance the reliability of the chip design and manufacturing process. Without EDA products, it would be too expensive, too time-consuming, or simply impossible for companies to design the complex ICs available in the market today.

30. The challenge that EDA software developers face is to design tools that provide solutions to new problems that arise in the EDA industry as demands increase for improved ICs. To understand these challenges, it is necessary to consider the characteristics of ICs that designers try to optimize when designing an IC.

1. Area

31. One IC characteristic that designers try to optimize is the area occupied by IC elements within a layout. In other words, it is desirable to minimize the area of an integrate

circuit layout. This desire is driven in part by the fact that ICs are fabricated on a silicon wafer (as shown in Figure F below), and it is desirable maximize the number of ICs that may be yielded from a wafer. Obviously, a larger number of ICs can be yielded from a wafer if the area of each IC is minimized.

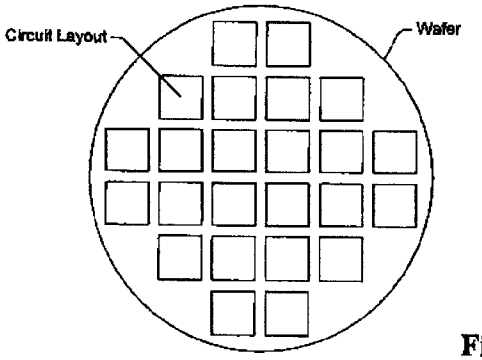

Figure F

32. Another reason for minimizing the area of a circuit layout is simply so that the chip package can be made smaller. IC's are commonly placed in packages, such as the one shown in Figure G, below. Electronics companies often prefer smaller chipsets because they can more easily fit into miniature electronic devices such as cell phones and digital cameras.

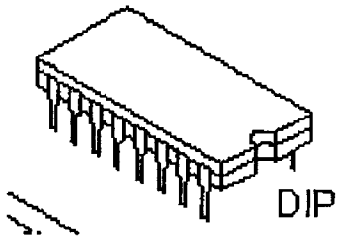

Figure G

2. Congestion

33. Another characteristic of ICs that designers try to optimize is called "congestion," which is a parameter that indicates the number of wires that cross a certain area of the layout. This IC characteristic is similar to the notion of congestion on a freeway, which increases as more cars drive along a certain stretch of the road. A circuit with too much congestion simply

cannot be routed because there is no room for the wires. Figure H, below, shows wires running horizontally across a portion of an IC layout. The upper part of the layout is badly congested, nearly completely full of wires. The lower portion is far less congested, and has room for many more wires to be routed. Because of the need to route wires between IC elements, designers try to minimize congestion.

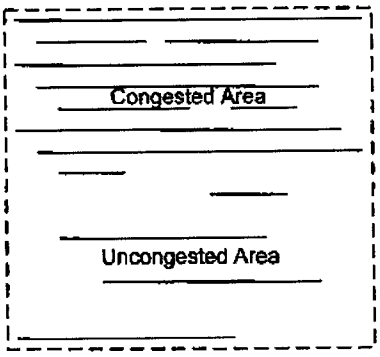

Figure H

3. Circuit Optimizations

34. To serve the needs of IC designers, the makers of EDA Tools (*e.g.*, Synopsys and Magma) are constantly trying to invent new features for their EDA tools that will allow the designers to achieve the goals of optimizing IC characteristics such as area and congestion. For example, EDA tools may include software processes for automatically selecting the smallest IC elements that can achieve the required function. As another example, EDA tools use algorithms for placing IC elements in a manner so that congestion is reduced.

35. Another way to optimize certain IC characteristics is by choosing the logic (*i.e.*, the types of gates, and their interconnections) for implementing the high-level circuit description. As already discussed, the logic shown in Figure C (reproduced below) can be chosen to implement the high-level circuit description of Equation (1), also reproduced below.

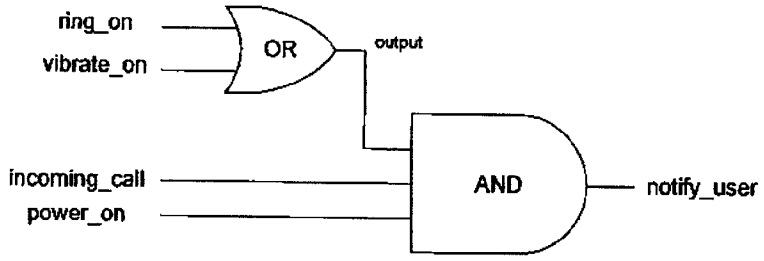

Figure C

notify_user=(ring_on OR vibrate_on) AND power_on AND incoming_call (1)

36. However, the logic shown in Figure C is not the only way to implement Equation (1). There are many combinations of different types of gates (*e.g.*, NAND gates, NOR gates, AND gates, OR gates) that may be chosen and interconnected to achieve the same function. As an example, during the logic synthesis stage, the high-level description of equation (1) may be transformed into a different netlist representing an alternative circuit, shown in Figure I, below.

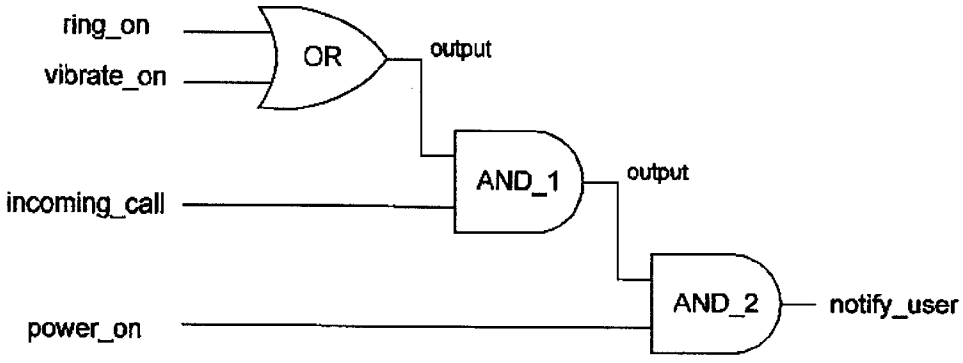

Figure I

37. The diagrams shown in Figures C and I show two different circuits for implementing the function specified by the description of Equation (1). The design shown in Figure H uses to two-input AND gates instead of the single three-input AND gate used in the design shown in Figure C. There are certain pros and cons to each design. For example, there is allot of congestion in the area proximate the inputs of the three-input AND gate of the design of Figure C. The design shown in Figure I eliminates the three-input AND gate (used in Figure C), and thus relieves congestion in one part of the layout. However, the use of two AND gates in Figure I (instead of one in Figure C) may require an increase in the overall area of the circuit

layout. Thus, it is apparent that great difficulty may be incurred in trying to optimize all of the different IC characteristics, especially when it is considered that a single IC may contain millions of gates.

V. OVERVIEW OF THE PATENTS AT ISSUE.

A. THE `508 PATENT.

38. The `508 patent provides a solution to relieve congestion by modifying the logic of the circuit (*i.e.*, "performing logic modifications"). Congestion occurs when too many wires need to be routed through too small of a space. There are two general ways that logic modification can relieve congestion. First, the logic modifications can reduce the number of wires in a portion (*i.e.*, an area or region) of the circuit, directly reducing congestion. *See, e.g.*, (1 A-6 at 2:36-48). The `508 Patent describes many examples of this type of logic modification. (1 A-7 at 4:22-5:44); Figures 3A-3C, 4A-4B, 5A-5B, and 6A -6B.

39. Second, the logic modifications can make the circuit faster. (1 A-6 at 2:36-48). If the circuit is faster, the circuit elements can be moved further apart while still meeting the speed requirements. Spreading the gates spreads the wires over a larger area, reducing congestion.

40. After the logic modifications are made, the placement is refined to reflect the modifications. If the logic modifications increase the area of the circuit elements too much, they will not fit in the existing placement. The placement will have to be redone, possibly resulting in congestion in different places. Therefore, logic modifications should not increase the area of the circuit elements excessively.

B. THE `745 PATENT.

41. IC designers often face the issue of capacitance in the wires of the integrated circuit. Capacitance causes the transmission of electrical signals through the wires to be delayed.

Capacitance occurs when wires are close together – the closer the wires, the greater the capacitance. It is important for IC designers to have a good estimate of how much capacitance is in a circuit so they can calculate the time it will take signals to travel from one part of the circuit to another.

42. The alleged invention of the '745 patent is a method of estimating capacitance based on the amount of congestion in the circuit. Integrated circuits are often divided conceptually into a grid of imaginary rectangular regions, which are used in placing and routing the circuit elements. When many wires run through the regions, the region is said to be congested, and the wires are packed closely together. This also means that the capacitance is large. Because capacitance is related to congestion, the patent says that the congestion can be used to estimate the capacitance.

VI. UNDERSTANDING OF CLAIM CONSTRUCTION PRINCIPLES.

43. I understand that the claims of a patent define the invention. It has been explained to me that the words of a claim are generally given their ordinary and customary meaning, which is the meaning that the term would have to a person of ordinary skill in the art. However, it has also been explained to me that the claims must be read in light of the specification, and be construed consistent with the specification. I also understand that the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess.

44. In addition to consulting the specification, I have considered the patent's prosecution history. I have been told that the prosecution history can sometimes inform the meaning of the claim language by demonstrating how the inventor understood the invention. In addition, I have also been told that the claims themselves provide substantial guidance as to the

meaning of particular claim terms. Finally, I understand that one may rely on extrinsic evidence in construing the claims where it is helpful.

45. It has been explained to me that an element in a claim may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. I understand that in construing means-plus-function claim limitations, one must first define the particular function claimed. Thereafter, one must identify "the corresponding structure, material, or acts described in the specification.

46. I understand that the function of a "means plus function" claim must be construed to include the limitations contained within the claim language. I have been told that the relevant function is usually recited in the claim after the prepositional link "for," which ties the means to the function. However, I have been instructed that, when it makes sense in light of the Specification, a "wherein" clause may further describe the claimed function.

47. It has also been explained to me that in a means-plus-function claim where the disclosed structure is a computer programmed to carry out an algorithm, the disclosed structure is not the general purpose computer, but rather the structure is a special purpose computer programmed to perform the disclosed algorithm. I understand that the disclosure of the algorithm in the specification may take many forms, such as a table, a flow chart or textually disclosed steps.

48. I also understand that the structure disclosed in the specification qualifies as "corresponding" structure only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim. Finally, I have been instructed that

distinct and alternative described structures may correspond with the claimed function if the Specification makes that clear.

VII. CONSTRUCTION OF CLAIM TERMS IN THE `508 PATENT.

A. "Means For Performing An Initial Placement"

49. Claims 17 and 18 of the `508 Patent both recite "means for performing an initial placement of integrated circuit elements within bins on the design layout." (1 A-9 at 8:34-35 and 8:48-50). I understand that Synopsys and Magma agree that the claimed function is "performing an initial placement of integrated circuit elements within bins on the design layout." AJCC, Exh. B, pp. 8-10. As I understand the situation, the only dispute with respect to this phrase concerns its disclosed corresponding structure. Specifically, Magma argues that there is no corresponding structure disclosed in the Specification. AJCC, Exh. B, pp. 8-10. On the other hand, Synopsys has identified the following structures corresponding with the "means for performing an initial placement of integrated circuit elements within bins on the design layout":

- an electronic design automation placement tool (1 A-7 at 3:30-31);
- a computer executing an algorithm for placing cells in one or more regions using a placement tool that partitions cells into one or more regions at each stage of the placement (1 A-7 at 3:31-35); and
- a computer executing an algorithm for placing cells in accordance with a placement algorithm that is limited by the topology of the circuit (1 A-7 at 4:23-29).

50. I disagree with Magma's assertion that there is no structure in the Specification of the `508 Patent that corresponds with the "means for performing an initial placement of integrated circuit elements within bins on the design layout." AJCC, Exh. B, pp. 8-10. The specification of the `508 Patent explains that the function of "performing an initial placement of integrated circuit elements within bins on the design layout" may be performed as follows:

> The present invention may be used in conjunction with an electronic design automation placement tool. In accordance with an exemplary embodiment of one such placement

tool, at each stage in cell placement, the cells are partitioned into a number of bins. (1 A-7 at 3:31-35).

51. A person of ordinary skill in the art would understand the statement that "the present invention may be used in conjunction with an electronic design automation placement tool" to mean that the claimed function of "performing an initial placement of integrated circuit elements within bins on the design layout" may be performed using any commercially available electronic design automation ("EDA") placement tool. At the time of the filing of the '508 Patent (*i.e.*, June 12, 1998), many EDA placement tools were available from various vendors including Cadence Design Systems and Avant! Corporation. Currently, Synopsys and Magma both sell EDA placement tools.

52. The Specification of the '508 Patent also explains that "[i]n accordance with an exemplary embodiment of one such placement tool, at each stage in cell placement, the cells are partitioned into a number of bins." (1 A-7 at 3:31-35). A person of ordinary skill in the art would understand this portion of the Specification as describing a well-known class of algorithms for performing the claimed function of "performing an initial placement of integrated circuit elements within bins on the design layout." For example, Michael Smith's well known text book on Application-Specific Integrated Circuits explains:

> The min-cut placement method uses successive application of partitioning [Breuer, 1977]. The following steps are shown in Figure 16.24:
>
> 1. Cut the placement area into two pieces.
> 2. Swap the logic cells to minimize the cut cost.
> 3. Repeat the process from step 1, cutting smaller pieces until all the logic cells are placed.
>
> Usually we divide the placement area into bins. The size of a bin can vary, from a bin size equal to the base cell (for a gate array) to a bin size that would hold several logic cells. We can start with a large bin size, to get a

rough placement, and then reduce the bin size to get a final placement.

(6 A-39 - A-40).

53. As reflected by the above citation, algorithms for "partitioning" may be distinct from the claimed function of "performing an initial placement of integrated circuit elements within bins on the design layout." Partitioning algorithms constitute just one of several different ways to perform this claimed function. For example, Michael Smith's text book also describes the "pairwise-interchange algorithm," which does not use partitioning. Instead, this method chooses a cell at random and evaluates the benefits of interchanging the cell's location with that of other cells being placed. (6 A-44).

54. The Specification of the `508 Patent describes a very specific algorithm where it states "**at each stage** in cell placement, the cells are partitioned into a number of bins." (1 A-7 at 3:31-35) (emphasis added). A person of ordinary skill in the art would understand this passage to describe an iterative "partitioning" process. This described algorithm is just one example of the broadly claimed function.

B. "Means for Calculating Congestion of the Initial Placement"

55. The phrase "means for calculating congestion of the initial placement" appears in claims 17 and 18 of the `508 Patent. (1 A-9 at 8:36 and 8:53). I understand that the parties agree that the claimed function is "calculating congestion of the initial placement." AJCC, Exh. B, pp. 7. Synopsys has proposed that the corresponding structure be construed as follows:

> (a) a computer executing an algorithm for calculating congestion for the initial placement in accordance with an algorithm that calculates the total number of pins in the bin divided by the total routable area in the bin (1 A-7 at 4:61-67); and
>
> (b) a computer executing an algorithm for calculating congestion for the initial placement using interconnection models for interconnects between bins or within bins (1 A-7 at 3:35-38).

56. I understand that Magma agrees that the first above-identified structure corresponds with the "means for calculating congestion of the initial placement." AJCC, Exh. B, pp. 7-8. Therefore, the only dispute I am aware of concerning this phrase is whether or not the second above-identified structure (hereinafter referred to as the "Interconnect Model Algorithm") also corresponds with this claim limitation.

57. The Specification of the `508 Patent links the description of the Interconnect Model Algorithm to the claimed function of "calculating congestion of the initial placement" where it states that "[i]nterconnection models for interconnects between bins and within bins provide both delay estimates for each interconnect in the circuit, as well as congestion estimates for each bin in the circuit." (1 A-7 at 3:35-38). This portion of the Specification conveys to a person of ordinary skill in the art that the claimed function of "calculating congestion of the initial placement" can be achieved using the Interconnect Model Algorithm. For example, global routing creates a model representing the approximate location of interconnects. This model can be used to compute congestion because it includes information on the approximate number of wires in a given region.

58. I understand that Magma has yet to provide any explanation for adopting only one of the two disclosed structures corresponding with the "means for calculating congestion of the initial placement." I reserve the right to present further testimony in response to any arguments or evidence Magma later identifies on this issue.

VIII. CONSTRUCTION OF CLAIM TERMS IN THE `745 PATENT.

59. An IC design consists of gates connected by wires. During the physical design process, an IC may be divided conceptually into areas called buckets. For simplicity and for the purposes of this declaration only, I will discuss wires that pass through a bucket. Because of the width and size of the wires, a bucket is only large enough to allow a certain number of wires to

pass through it.

60. The '745 patent at 8:33-36 states the following:

> "The congestion score for a bucket is defined as the ratio of the routing resources used so far to the total routing resources available in the bucket." (8 A-67 at 8:33-36).

61. The definition at 8:33-36 says that a congestion score is "the routing resources used so far" divided by "the total routing resources available." The "routing resources" are wires that run through the bucket. The "total routing resources available" are the number of wiring spaces or tracks available for wires in the bucket. (8 A-67 at 8:30-44). The congestion score will be a number between 0 and 1. A congestion score of 0 means there are no wires in the bucket, and a congestion number of 1 means you have as many wires as there are spaces or tracks for wires. In other words, a large congestion score (one that is close to 1) means a lot of wires, and the circuit is more congested.

62. The definition at 8:33-36 also comports with the ordinary, common-sense understanding of the word "congestion." To "congest" means "[t]o overfill or overcrowd." (10 A-79). "Congestion" therefore refers to the degree to which a space is filled or crowded. Greater congestion means more filling and more crowding, not less.

63. The definition at 8:33-36 is also consistent with the rest of the disclosure of the specification (with the exception of claim 2). For example, this definition is consistent with Figure 7 and the text describing this figure at 8:30-44. Figure 7 shows a bucket that has three wires running through the bucket in the vertical direction and has four total vertical wiring spaces available. The congestion score is given as 0.75 – that is, 3 divided by 4. It is also consistent with other parts of the patent that deal with measuring congestion score, including 8:45-9:29.

64. The use of the term "congestion score" in claim 2, on the other hand, is not

consistent with the definition at 8:33-36 or the rest of the specification. Claim 2 recites the following:

> [t]he method of claim 1, wherein the congestion score is a ratio of a number of available wire routing spaces in a given layer of the bucket in a given direction to a total number of wire routing spaces in the given layer of the bucket in the given direction."

65. That is, a congestion score is the number of open, available spaces or tracks for wires to pass through (*i.e.*, "available wire routing spaces") divided by the total number of wires that could possibly pass through (*i.e.*, "total number of wire routing spaces"). Using the claim 2 language, a large congestion score means fewer wires, and thus less congestion, because if there are a lot of open, available spaces for wires, the number of wires must be few. That is, the quantity called "congestion score" in claim 2 is actually a measure of the lack of congestion. This is the opposite of the definition in the specification.

66. This use of the term "congestion score" is also inconsistent with the use of the term elsewhere in the specification. For example, using the claim 2 formulation of "congestion score," the congestion score for the vertical direction in Figure 7 would be 0.25, not 0.75, because that is the number of "available wire routing spaces" – 1 – divided by the "total number of wire routing spaces" – 4. In fact, the patent specification nowhere discloses a "congestion score" as the term is used in claim 2.

67. This use of the term "congestion score" is also inconsistent with the ordinary meaning of the word "congestion." In the example of Figure 7, a bucket that is three-fourths filled would have a congestion score of 0.25 (or 25%) as the term is used in claim 2. A bucket that is completely empty would have a congestion score of 1 (or 100%). Again, this is not a measure of congestion, but a measure of lack of congestion.

68. In summary, claim 2 uses the term "congestion score" in a way that is inconsistent

with the patent definition provided at 8:33-36, inconsistent with the rest of the patent specification, and inconsistent with the ordinary meaning of the term "congestion."

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct.

EXECUTED on 3 November, 2006

David Harris

David Harris, Ph.D.

EXHIBIT A

DR. DAVID MONEY HARRIS

301 Platt Blvd.
Claremont, CA 91711
(909) 607-3623
David_Harris@hmc.edu
http://www3.hmc.edu/~harris

Professional Interests: High-speed digital VLSI.
Microprocessor design, computer architecture, integer & floating point execution units, graphics, communications, networks, high speed circuits, clocking, embedded systems, security, wireless systems.

Education:

Stanford University **Stanford, CA**
Ph.D. March 1999. *Skew-Tolerant Domino Circuits* with Mark Horowitz.
GPA: 4.0 / 4.0

Massachusetts Institute of Technology **Cambridge, MA**
M. Eng. Degree in Electrical Engineering and Computer Science,
S. B. Degree in Electrical Engineering, S.B. Degree in Mathematics,
May 1994. **GPA: 5.0 / 5.0**

Experience:

Harvey Mudd College **Claremont, CA**
1999-Present. Associate Professor of Engineering.

Harris Consulting **Upland, CA**
2004-Present. Testifying expert at Dechert LLP in Synopsys v. Magma case. Patent claims construction, conception, ownership, and infringement analysis, validity and prior art, code analysis, expert reports, depositions, and testifying in federal court. Specialty in integrated circuits and electronic design automation.

One Hot Logic **Upland, CA**
2001-2005. President and Founder. Profitable startup manufacturing low-cost functional chip testers.

Sun Microsystems Labs **Mountain View, CA**
1997-Present. Visiting Professor. Research on low-overhead asynchronous applications of domino circuits with Turing award-winner Ivan Sutherland. Consulting one day per week. Multiple patents.

Hewlett Packard **Fort Collins, CO**
May-August 2000. Consultant. Developed clock skew budgets and power grid analysis tools for Itanium 2 microprocessor. Taught circuit design seminars.

Evans & Sutherland **Salt Lake City, UT**
June-August 1999. Consultant. Architected OpenGL Geometry Engine delivering 44 Mvertices/s.

Intel Corporation **Santa Clara, CA**
1994-1997. Logic designer for Itanium microprocessor; designed Integer Execution and Multimedia units, coded and tested several units. Circuit design on Pentium II & Itanium microprocessors. Trained

junior logic and circuit designers. Multiple patents.
May-August 2004. Visiting Professor. Reconfigurable cryptographic Accelerators.

Design experience with Cadence, HSPICE, Verilog, Synopsys, Pathmill, FPGAs, and embedded microcontrollers. C, Perl, Java, etc.

Consulting experience with HAL Computer, Rockwell Semiconductor, TRW, Circuit Semantics, LightTime, Multigig.

Honors & Activities:

National Science Foundation fellow, winner of Microsoft Technical fellowship, Irwin Sizer Award for Most Significant Improvement to MIT Education, Putnam Exam honors, Tau Beta Pi member, 6.004 Computer Optimization Contest Winner, Secretary of Educational Studies Program, M. Eng. Thesis Prize recipient, President of Intel Newhire Network, founder and Chairperson of Stanford Educational Studies Program, Lyon's Award for Public Service, Commercial Pilot License, AngelFlight volunteer pilot, ISSCC Intl. Technical Program Committee.

Patents:

US Patent 6,995,039: Method and Apparatus for Electrostatically Aligning Integrated Circuits. Feb. 7, 2006.
US Patent 6,970,034: Method and Apparatus for Reducing Power Consumption During Sleep Mode. Nov. 29, 2005.
US Patent 6,769,007: Adder Circuit with a Regular Structure. July 27, 2004.
US Patent 6,710,436: Method and Apparatus for Electrostatically Aligning Integrated Circuits. March 23, 2004.
US Patent 6,580,303: Datapath Control Circuit with Adjustable Delay Elements. June 17, 2003.
US Patent 6,239,622: Self-timed Domino Circuit. May 29, 2001.
US Patent 6,169,422: Apparatus and Methods for High Throughput Self-Timed Domino Circuits. Jan 2, 2001.
US Patent 5,880,985: Efficient Combined Array for 2n bit n bit Multiplications. March 9, 1999.
US Patent 5,880,608: Pulsed Domino Latches. March 9, 1999.
US Patent 5,821,775: Interface Between Monotonic and Nonmonotonic Logic. October 13, 1998.
US Patent 5,517,136: Opportunistic Time-Borrowing Domino Logic. May 14, 1996.

Other patents pending in the area of high-speed circuit/logic design.

Publications:

Harris, David, and Harris, Sarah, Digital Design and Computer Architecture, Morgan Kaufmann Publishers, to be published 2007.

Jiang, Nan, and Harris, David, "Quotient Pipelined Very High Radix Scalable Montgomery Multipliers, *Asilomar Conf. Signals, Systems, and Computers*, Nov. 2006, pp. ***.

Harris, David, and Harris, Sarah, "From Zero to One: An Introduction to Digital Design and Computer Architecture," *Reconfigurable Computing in Education*, March 2006.

John Robinson with David Harris, *San Bernardino Mountain Trails*, Wilderness Press, 2006.

Utter, Alex, Chen, Henry, Ouchi, Shane, Harris, David, Rainer, Amanda, Kaneakua, Keane, Prounh, Chris, and Osofsky, Samuel, "Adaptive Two-Channel Automatic Gain Control System," *IEEE Aerospace Conference*, March 2006, pp. 1-8.

Kelley, Kyle, and Harris, David, "Parallelized Very High Radix Scalable Montgomery Multipliers," *Asilomar Conf. Signals, Systems, and Computers*, Nov. 2005, pp. 1196-1200.

Chen, H., Shi, R., Cheng, C.K., and Harris, David, "Surfliner: A Distortionless Electrical Signaling Scheme for Speed of Light On-Chip Communications," *IEEE Intl. Conf. on Computer Design*, 2005, pp. 497-502.

Kelley, Kyle, and Harris, David, "Very High Radix Scalable Montgomery Multipliers," *5th Intl. Workshop on System-on-Chip*, July 2005, pp. 400-404.

Harris, Sarah, and Harris, David, "Inexpensive Student-Assembled FPGA / Microcontroller Board," *Microelectronics Systems Education Conf.*, June 2005, pp. 101-102. (Best Poster Award.)

Harris, David, Krishnamurthy, Ram, Anders, Mark, Mathew, Sanu, and Hsu, Steven, "An Improved Unified Scalable Radix-2 Montgomery Multiplier," *IEEE Symposium on Computer Arithmetic*, June 2005, pp. 172-178.

Harris, David, "Logical Effort of Higher Valency Adders," *Asilomar Conf. Signals, Systems, and Computers*, Nov. 2004, pp. 1358-1362.

Linderman, Michael, Harris, David, and Diaz, David, "Bounding Bus Delay and Noise Effects of On-Chip Inductance," *IEEE Workshop on Signal Propagation on Interconnects*, April 2004.

Weste, Neil, and Harris, David, *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison Wesley, May 2004.

Harris, David, "An Exponentiation Unit for an OpenGL Lighting Engine," *IEEE Trans. Computers*, vol. 53, no. 3, March 2004, pp. 251-258.

Harris, David, "A Taxonomy of Prefix Networks," *Asilomar Conf. Signals, Systems, and Computers*, Nov. 2003, pp. 2213-2217.

Harris, David, and Sutherland, Ivan, "Logical Effort of Prefix Adders,"

Asilomar Conf. Signals, Systems, and Computers, Nov. 2003, pp. 673-678.

Harris, David, and Akin, Tayfun, "A Cross-Cultural Chip Design Project," *Proc. American Society of Engineering Educators Annual Conf.*, June 2003.

Harris, David, and Diaz, David, "TestosterICs: A Low-Cost Functional Chip Tester," *Microelectronics Systems Education Conf.*, June 2003.

Harris, David, Breed, Genevive, Erler, Matthew, and Diaz, David, "Comparison of Noise Tolerant Precharge (NTP) to Conventional Feedback Keepers for Dynamic Logic," *Great Lakes Symposium on VLSI*, April 2003.

Harris, David, "The Microprocessor as a Microcosm: A Hands-on Approach to VLSI Design Education," *ASEE/IEEE Frontiers in Education Conf.*, November 2002.

Harris, David, "A Freshman Advising Seminar on Digital Electronics and Chip Design," *Proc. American Society of Engineering Educators Annual Conf.*, June 2002.

Harris, David, and Naffziger, Sam, "Statistical Clock Skew Modeling with Data Delay Variations," *IEEE Trans. VLSI Systems*, vol. 9, no. 6, December 2001, pp. 888-898.

Harris, David, "A Powering Unit for an OpenGL Lighting Engine," *Proc. Asilomar Conf. on Signals, Systems, and Computers*, November 2001. (invited paper)

Harris, David, "A Case for Project-Based Design Education," *Intl. J. Engineering Education*, vol. 17, no 4-5, pp. 367-369, 2001.

Harris, David, "Skew-Tolerant Domino Circuits," *Tech-Online Online Symposium for Electronics Engineers*, Fall 2000. www.techonline.com

Harris, David, *Skew-Tolerant Circuit Design*, Morgan Kaufmann Publishers, May 2000.

Harris, David, Horowitz, Mark, and Liu, Dean, "Timing Analysis Including Clock Skew," *IEEE Trans. Computer-Aided Design*, November 1999, pp. 1608-1618.

Harris, David, "A Case for Project-Based Design Education," *Proc. Mudd Design Workshop II*, Claremont, CA, May 1999.

Coats, Bill, *et al.*, "A Counterflow pipeline experiment," *Proc. Fifth Intl. Conf. Asynchronous Circuits and Systems*, Barcelona, Spain, April 1999, pp. 161-172.

Harris, David, Horowitz, Mark, and Liu, Dean, "Timing Analysis Including Clock Skew," *International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Monterey, CA, March 1999, pp. 15-20.

Sutherland, Ivan, Sproull, Bob, and Harris, David, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann Publishers, 1999.

Harris, David, and Horowitz, Mark, "Skew-Tolerant Domino Circuits," *IEEE J. Solid-State Circuits*, Nov. 1997, pp. 1702-1711.

Harris, David, Oberman, Stuart, and Horowitz, Mark, "SRT Division Algorithms and Architectures," *Proc. 13th IEEE Symposium on Computer Arithmetic*, July 1997.

Harris, David, and Horowitz, Mark, "Skew-Tolerant Domino Circuits," *ISSCC Dig. Tech Papers*, Feb. 1997, pages 422-423.

Dally, W., Dennison, L., Harris, D., Kan, K., and Xanthopoulos, T., "The Reliable Router: A Reliable and High-Performance Communication Substrate for Computers," *Lecture Notes in Computer Science, Vol. 853, Proc. First Intl. Workshop on Parallel Computer Routing and Communication*, pp. 241-255, 1994.

Dally, W., Dennison, L., Harris, D., Kan, K., and Xanthopoulos, T., "Architecture and implementation of the reliable router," *Hot Interconnects II*, 1994.

Barbieri, Harris, & Eeckman, "A Simulation of Neural Processing in the Auditory Path of the Barn Owl," in Eeckman (ed) *Computation in Neurons and Neural Systems*; Kluwer Academic Publishers, 1994.

Harris, David, and Harris, Daniel "A Low-Cost pH Meter for the Classroom," *J. Chem. Educ.* 1992, Volume 69, page 563.

Harris, David "A Taxonomy of Ceiling Tile," *J. Irreproducible Results*, 1992, Volume 37, Number 3, page 7-8.

Invited Talks

David Evans Conf. Computer Eng. 2005 CMOS VLSI Design
DATE 2004 Advanced Domino Circuit Design Tutorial
ASPDAC 2003 High-Speed CMOS Circuit Design Tutorial
ISSCC 2002 Microprocessor Design Workshop
ISSCC 2001 Logical Effort Tutorial
ISSCC 2001 Panel Session on Taming the Microprocessor Monster
ISSCC 2000 Workshop on High Speed Microprocessor Design

Teaching Experience:

E155: Microprocessor Applications
Harvey Mudd College Fall 2006, 2004, 2003, 2002, 2000, 1999
E151: Engineering Electronics

Harvey Mudd College Fall 2001
E59: Introduction to Systems Engineering
Harvey Mudd College Fall 2004, 2002, 2001
E158: Introduction to CMOS VLSI Design
Harvey Mudd College Spring 2005, 2004, 2003, 2002, 2001
FYS1: Digital Electronics & Chip Design for Freshmen
Harvey Mudd College Fall 2006, 2004, 2003, 2002, 2001, 2000, 1999
E85: Introduction to Computer Engineering
Harvey Mudd College Spring 2001, 2000, 1999
E101: Advanced Systems Engineering
Harvey Mudd College Fall 2003
Engineering Clinic
Harvey Mudd College
Intel Corporation Power Grid Analysis 2003
Sandia National Labs FT Ion Mobility Spectrometer 2003, 2002
Qualcomm GPS Data Recorder 2002
Aerospace Corporation Bit Error Rate Tester 2001, A/D 2004
Texas Instruments GPS Searcher 2000, 2001
Opto22 I/O Controller 2000
Sun Microsystems Chip Tester 1998, 1999
Evans & Sutherland Geometry Engine 1999
Sierra Wireless USB / Ethernet Bridge 2004
Southwest Research Institute 2006
MaxViz 2006
High Speed CMOS VLSI Design
Industrial courses taught at:
Sun Microsystems Winter 2004
Qualcomm Summer 2003
Evans & Sutherland Summer 1999
Intel Corporation Fall 1998
Intel Corporation Spring 1998
UC Berkeley Extension Winter 1998
HAL Computer Fall 1997
EE371: Advanced VLSI Circuit Design
Teaching Fellow (co-taught with Professor M. Horowitz): Stanford 1996
Digital Electronics & Chip Design (for high school students)
Educational Studies Program: Stanford 1996-1998
EE271: Introduction to VLSI Design
Teaching Assistant for Professor M. Horowitz: Stanford 1995
Advanced VLSI Design (intensive course at Cray Computer)
Teaching Assistant for Professor W. Dally: 1994
6.090: "VLSI for Freshmen and Sophomores:" master's thesis project
Instructor (co-taught with Professor W. Dally): MIT 1994
6A27: Introduction to Digital Electronics (freshman seminar)
Instructor: MIT 1993-1994
18.06: Linear Algebra & 18:03: Differential Equations
Tutor: MIT Experimental Study Group 1991-1994

Academic Leadership:
Director, Engineering Computational Facility 2001-
Chair, Engineering Faculty Search Committee 2001, 2004-05
Chair, Engineering Petitions Committee, 2000-01, 2003, 2006-07

Chair, Scholarly Standing Committee, 2004-05, 2006-07
Co-chair, Strategic Planning Committee, 2006-07

Grants:

2006: Jay Wolkin '99 and Clay Family Foundation Fellowship: $500,000
2005-07: Intel Corporation, "High Performance and Low-Power Reconfigurable Circuits for Cryptography Accelerators," $30,000